# Performance Management Commands

This chapter describes the commands used to manage router performance on the network. To manage system performance, you can perform load-balancing and modify system parameter. For example, priority queueing allows you to prioritize traffic order.

See the *Internetwork Design Guide* for additional information.

For performance management configuration tasks and examples, refer to the chapter entitled "Managing System Performance" in the *Configuration Fundamentals Configuration Guide*.

# buffers

Use the **buffers** global configuration command to make adjustments to initial buffer pool settings and to the limits at which temporary buffers are created and destroyed. Use the **no** form of this command to return the buffers to their default size.

**buffers** {**small** | **middle** | **big** | **verybig** | **large** | **huge** | *type number*} {**permanent** | **max-free** | **min-free** | **initial**} *number*
**no buffers** {**small** | **middle** | **big** | **verybig** | **large** | **huge** | *type number*} {**permanent** | **max-free** | **min-free** | **initial**} *number*

## Syntax Description

| | |
|---|---|
| **small** | Buffer size of this public buffer pool is 104 bytes. |
| **middle** | Buffer size of this public buffer pool is 600 bytes. |
| **big** | Buffer size of this public buffer pool is 1524 bytes. |
| **verybig** | Buffer size of this public buffer pool is 4520 bytes. |
| **large** | Buffer size of this public buffer pool is 5024 bytes. |
| **huge** | Default buffer size of this public buffer pool is 18024 bytes. This value can be configured with the **buffers huge size** command. |
| *type number* | Interface type and interface number of the interface buffer pool. The type value cannot be **fddi**. |
| **permanent** | Number of permanent buffers that the system tries to create and keep. Permanent buffers are normally not trimmed by the system. |
| **max-free** | Maximum number of free or unallocated buffers in a buffer pool. A maximum of 20,480 small buffers can be constructed in the pool. |
| **min-free** | Minimum number of free or unallocated buffers in a buffer pool. |
| **initial** | Number of additional temporary buffers that are to be allocated when the system is reloaded. This keyword can be used to ensure that the system has necessary buffers immediately after reloading in a high-traffic environment. |
| *number* | Number of buffers to be allocated. |

## Default

The default number of buffers in a pool is determined by the hardware configuration and can be displayed with the EXEC **show buffers** command.

## Command Mode

Global configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

Normally you need not adjust these parameters; do so only after consulting with technical support personnel. Improper settings can adversely impact system performance.

You cannot configure FDDI buffers.

## Examples of Public Buffer Pool Tuning

In the following example, the system will try to keep at least 50 small buffers free:

```
buffers small min-free 50
```

In the following example, the permanent buffer pool allocation for big buffers is increased to 200:

```
buffers big permanent 200
```

## Example of Interface Buffer Pool Tuning

A general guideline is to display buffers with the **show buffers** command, observe which buffer pool is depleted, and increase that one.

In the following example, the permanent Ethernet 0 interface buffer pool on a Cisco 4000 is increased to 96 because the Ethernet 0 buffer pool is depleted:

```
buffers ethernet 0 permanent 96
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**buffers huge size**
**show buffers**

# buffers huge size

Use the **buffers huge size** global configuration command to dynamically resize all huge buffers to the value you specify. Use the **no** form of this command to restore the default buffer values.

**buffers huge size** *number*
**no buffers huge size** *number*

## Syntax Description

*number* Huge buffer size, in bytes.

## Default

18024 bytes

## Command Mode

Global configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

Use only after consulting with technical support personnel. The buffer size cannot be lowered below the default.

## Example

In the following example, the system will resize huge buffers to 20000 bytes:

```
buffers huge size 20000
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**buffers**
**show buffers**

# custom-queue-list

To assign a custom queue list to an interface, use the **custom-queue-list** interface configuration command. To remove a specific list or all list assignments, use the **no** form of the command.

**custom-queue-list** *list*
**no custom-queue-list** [*list*]

## Syntax Description

| | |
|---|---|
| *list* | Number of the custom queue list you want to assign to the interface. An integer from 1 to 16. |

## Default

No custom queue list is assigned.

## Command Mode

Interface configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

**Note** Custom queuing is not supported on tunnels.

Only one queue list can be assigned per interface. Use this command in place of the **priority-list** command (not in addition to it). Custom queueing allows a fairness not provided with priority queueing. With custom queueing, you can control the interfaces' available bandwidth when it is unable to accommodate the aggregate traffic enqueued. Associated with each output queue is a configurable byte count, which specifies how many bytes of data should be delivered from the current queue by the system before the system moves on to the next queue. When a particular queue is being processed, packets are sent until the number of bytes sent exceeds the queue byte count or until the queue is empty.

Use the **show queueing custom** and **show interface** commands to display the current status of the custom output queues.

## Example

In the following example, custom queue list number 3 is assigned to serial interface 0:

```
interface serial 0
custom-queue-list 3
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**queue-list default**
**queue-list interface**
**queue-list protocol**
**queue-list queue byte-count**
**queue-list queue limit**

# fair-queue

To enable weighted fair queueing for an interface, use the **fair-queue** interface configuration command. To disable weighted fair queuing for an interface, use the **no** form of this command.

**fair-queue** [*congestive-discard-threshold* [*dynamic-queues* [*reservable-queues*]]]
**no fair-queue**

## Syntax Description

| | |
|---|---|
| *congestive-discard-threshold* | (Optional) Number of messages allowed in each queue in the range 1 to 4096. The default is 64 messages. When the number of messages in the queue for a high-bandwidth conversation reaches the specified threshold, new high-bandwidth messages are discarded. |
| *dynamic-queues* | (Optional) Number of dynamic queues used for best-effort conversations (that is, a normal conversation not requiring any special network services). Values are 16, 32, 64, 128, 256, 512, 1024, 2048, and 4096. The default is 256. |
| *reservable-queues* | (Optional) Number of reservable queues used for reserved conversations in the range 0 to 1000. The default is 0. Reservable queues are used for interfaces configured for the Resource Reservation Protocol (RSVP) feature. |

## Default

Fair queuing is enabled by default for physical interfaces whose bandwidth is less than or equal to 2.048 megabits per second (Mbps) and that do not use Link Access Procedure, Balanced (LAPB), X.25, or Synchronous Data Link Control (SDLC) encapsulations. (Fair queuing is not an option for these protocols.) However, if custom queuing or priority queuing is enabled for a qualifying link, it overrides fair queueing, effectively disabling it. Additionally, fair queuing is automatically disabled if you enable autonomous or SSE switching.

Fair queueing is now enabled automatically on interfaces configured for Multilink PPP.

Congestive-discard-threshold: 64 messages; dynamic-queues: 256; reservable-queues: 0.

## Command Mode

Interface configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 11.0.

Fair queuing is supported for all LAN and line (WAN) protocols except X.25. These protocols are listed in “Default.” Fair queuing is on by default for interfaces with bandwidth less than or equal to 2 Mbps.

**Note** Fair queuing is not supported on tunnels.

When enabled for an interface, weighted fair queueing provides traffic priority management that automatically sorts among individual traffic streams without requiring that you first define access lists. Enabling weighted fair queueing requires use of this command only.

Weighted fair queuing can manage duplex data streams, such as those between pairs of applications, and simplex data streams such as voice or video. From the perspective of weighted fair queueing, there are two categories of sessions: high-bandwidth sessions and low-bandwidth sessions. Low-bandwidth traffic has effective priority over high-bandwidth traffic, and high-bandwidth traffic shares the transmission service proportionally according to assigned weights.

When weighted fair queuing is enabled for an interface, new messages for high-bandwidth traffic streams are discarded after the configured or default congestive-messages threshold has been met. However, low-bandwidth conversations, which include control-message conversations, continue to enqueue data. As a result, the fair queue may occasionally contain more messages than its configured threshold number specifies.

Weighted fair queuing uses a traffic data stream discrimination registry service to determine which traffic stream a message belongs to. For each forwarding protocol, Table 78 shows the attributes of a message that are used to classify traffic into data streams.

**Table 78 Weighted Fair Queuing Traffic Stream Discrimination Fields**

| Forwarder | Fields Used |
|---|---|
| AppleTalk | • Source net, node, socket<br>• Destination net, node, socket<br>• Type |
| CLNS | • Source NSAP<br>• Destination NSAP |
| DECnet | • Source address<br>• Destination address |
| Frame Relay switching | • DLCI value |
| DDN IP | • TOS<br>• IP Protocol<br>• Source IP address (if message is not fragmented)<br>• Destination IP address (if message is not fragmented)<br>• Source TCP/UDP port<br>• Destination TCP/UDP port |
| Transparent bridging | • Unicast: Source MAC, Destination MAC<br>• Ethertype SAP/SNAP multicast: Destination MAC address |
| Source-route bridging | • Unicast: Source MAC, Destination MAC<br>• SAP/SNAP multicast: Destination MAC address |
| VINES | • Source Network/Host<br>• Destination Network/Host<br>• Level 2 Protocol |
| Apollo | • Source Network/Host/Socket<br>• Destination Network/Host/Socket<br>• Level 2 protocol |

**Table 78 Weighted Fair Queuing Traffic Stream Discrimination Fields (Continued)**

| Forwarder | Fields Used |
|---|---|
| XNS | • Source/Destination Network/Host/Socket<br>• Level 2 Protocol |
| Novell NetWare | • Source/Destination Network/Host/Socket<br>• Level 2 Protocol |
| All others (default) | Control protocols (one queue per protocol) |

It is important to note that IP precedence, congestion in Frame Relay switching, and discard eligibility flags affect the weights used for queuing.

IP precedence, which is set by the host or by policy maps, is a number in the range of 0 to 7. Data streams of precedence *number* are weighted so that they are given an effective bit rate of *number*+1 times as fast as a data stream of precedence 0, which is normal.

In Frame Relay switching, message flags for congestion (FECN and BECN) and discard eligible (DE) message flags cause the algorithm to select weights that effectively impose reduced queue priority, providing the application with "slow down" feedback and sorting traffic, giving the best service to applications within their Committed Information Rate.

**Note** For Release 10.3 and earlier for the Cisco 7000 and 7500 with an RSP card, if you used the **tx-queue-limit** command to set the transmit (tx-queue) limit available to an interface on an MCI or SCI card and you configured custom queuing or priority queuing for that interface, the configured transmit (tx-queue) limit was automatically overridden and set to 1. With this release, for weighted fair queuing, custom queuing, and priority queuing, the transmit (tx-queue) limit is derived from the bandwidth value set for the interface using the bandwidth command. Bandwidth value divided by 512 rounded up yields the effective transmit (tx-queue) limit. However, the derived value only applies in the absence of a **tx-queue-limit** command; that is, a configured transmit (tx-queue) limit overrides this derivation.

When Resource Reservation Protocol (RSVP) is configured on an interface that supports fair queuing or on an interface that is configured for fair queuing with the reservable queues set to 0 (the default), the reservable queue size is automatically configured using the following method: interface bandwidth divided by 32 kbps. You can override this be specifying a reservable queue other than 0. For more information on RSVP, refer to the "Configuring RSVP" chapter in the *Network Protocols Configuration Guide, Part 1*.

## Examples

The following example enables use of weighted fair queuing on Serial 0, with a congestive threshold of 300. This means that messages will be discarded from the queuing system only when 300 or more messages have been queued and the message is in a data stream that has more than one message in the queue. The transmit queue limit is set to 2, based on the 384-kilobit (kb) line set by the bandwidth command:

```
interface serial 0
 bandwidth 384
 fair-queue 300
```

The following example requests a fair queue with 512 dynamic queues, 18 RSVP queues, and a congestive discard threshold of 64 messages:

```
interface Serial 3/0
 ip unnumbered Ethernet 0/0
 fair-queue 64 512 18
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**custom-queue-list**
**ip rsvp bandwidth**
**priority-group**
**priority-list default**
**queue-list default**
**random-detect**
**show interface**

# load-interval

To change the length of time for which data is used to compute load statistics, use the **load-interval** interface configuration command. Use the **no** form of this command to revert to the default setting.

**load-interval** *seconds*
**no load-interval** *seconds*

## Syntax Description

| | |
|---|---|
| *seconds* | Length of time for which data is used to compute load statistics. A value that is a multiple of 30, from 30 to 600 (30, 60, 90, 120, and so forth). |

## Default

300 seconds (or 5 minutes)

## Command Mode

Interface configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.3.

If you want load computations to be more reactive to short bursts of traffic, rather than averaged over 5-minute periods, you can shorten the length of time over which load averages are computed.

If the load interval is set to 30 seconds, new data is used for load calculations over a 30-second period. This data is used to compute load statistics, including input rate in bits and packets per second, output rate in bits and packets per second, load, and reliability.

Load data is gathered every 5 seconds. This data is used for a weighted average calculation in which more-recent load data has more weight in the computation than older load data. If the load interval is set to 30 seconds, the average is computed for the last 30 seconds of load data.

The **load-interval** command allows you to change the default interval of 5 minutes to a shorter or longer period of time. If you change it to a shorter period of time, the input and output statistics that are displayed when you use the **show interface** command will be more current, and based on more instantaneous data, rather than reflecting a more average load over a longer period of time.

This command is often used for dial backup purposes, to increase or decrease the likelihood of a backup interface being implemented, but it can be used on any interface.

## Example

In the following example, the default 5-minute average is set it to a 30-second average. A burst in traffic that would not trigger a dial backup for an interface configured with the default 5-minute interval might trigger a dial backup for this interface that is set for a shorter, 30-second interval.

```
interface serial 0
 load-interval 30
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**show interfaces**

# priority-group

To assign the specified priority list to an interface, use the **priority-group** interface configuration command. Use the **no** form of this command to remove the specified priority group assignment.

**priority-group** *list*
**no priority-group**

## Syntax Description

| | |
|---|---|
| *list* | Priority list number assigned to the interface. An integer from 1 to 16. |

## Default

None

## Command Mode

Interface configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

**Note** Priority queuing is not supported on tunnels.

Only one list can be assigned per interface. Priority output queueing provides a mechanism to prioritize packets transmitted on an interface.

Use the **show queuing priority** and **show interface** commands to display the current status of the output queues.

## Example

The following example causes packets on serial interface 0 to be classified by priority list 1:

```
interface serial 0
 priority-group 1
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**priority-list default**
**priority-list interface**
**priority-list queue-limit**
**queue-list default**

# priority-list default

To assign a priority queue for those packets that do not match any other rule in the priority list, use the **priority-list default** global configuration command. Use the **no** form of this command to return to the default or assign **normal** as the default.

**priority-list** *list-number* **default** {**high** | **medium** | **normal** | **low**}
**no priority-list** *list-number* **default** {**high** | **medium** | **normal** | **low**}

## Syntax Description

| | |
|---|---|
| *list-number* | Arbitrary integer between 1 and 16 that identifies the priority list selected by the user. |
| **high** \| **medium** \| **normal** \| **low** | Priority queue level. |

## Default

The **normal** queue, if you use the **no** form of the command.

## Command Mode

Global configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

When using multiple rules, remember that the system reads the priority settings in order of appearance. When classifying a packet, the system searches the list of rules specified by **priority-list** commands for a matching protocol or interface type. When a match is found, the packet is assigned to the appropriate queue. The list is searched in the order it is specified, and the first matching rule terminates the search.

## Example

The following example sets the priority queue for those packets that do not match any other rule in the priority list to a low priority:

```
priority-list 1 default low
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**priority-group**
**show queueing**

# priority-list interface

To establish queuing priorities on packets entering from a given interface, use the **priority-list interface** global configuration command. Use the **no** form of this command with the appropriate arguments to remove an entry from the list.

**priority-list** *list-number* **interface** *interface-type interface-number* {**high** | **medium** | **normal** | **low}**
**no priority-list** *list-number* **interface** *interface-type interface-number* {**high** | **medium** | **normal** | **low**}

## Syntax Description

| | |
|---|---|
| *list-number* | Arbitrary integer between 1 and 16 that identifies the priority list selected by the user. |
| *interface-type* | Specifies the name of the interface. |
| *interface-number* | Number of the specified interface. |
| **high** \| **medium** \| **normal** \| **low** | Priority queue level. |

## Default

No queuing priorities are established.

## Command Mode

Global configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

When using multiple rules, remember that the system reads the priority settings in order of appearance. When classifying a packet, the system searches the list of rules specified by **priority-list** commands for a matching protocol or interface type. When a match is found, the packet is assigned to the appropriate queue. The list is searched in the order it is specified, and the first matching rule terminates the search.

## Example

The following example sets any packet type entering on Ethernet interface 0 to a medium priority:

```
priority-list 3 interface ethernet 0 medium
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**priority-group**
**show queueing**

# priority-list protocol

To establish queuing priorities based upon the protocol type, use the **priority-list protocol** global configuration command. Use the **no** form of this command with the appropriate list number to remove an entry from the list.

**priority-list** *list-number* **protocol** *protocol-name* {**high** | **medium** | **normal** | **low**} *queue-keyword keyword-value*
**no priority-list** *list-number* **protocol** [*protocol-name* {**high** | **medium** | **normal** | **low**} *queue-keyword keyword-value*]

## Syntax Description

| | |
|---|---|
| *list-number* | Arbitrary integer between 1 and 16 that identifies the priority list selected by the user. |
| *protocol-name* | Specifies the protocol type: **aarp**, **arp**, **apollo**, **appletalk**, **bridge** (transparent), **clns**, **clns_es**, **clns_is**, **compressedtcp**, **cmns**, **decnet**, **decnet_node**, **decnet_router-l1**, **decnet_router-l2, dlsw**, **ip**, **ipx**, **pad**, **rsrb**, **stun**, **vines**, **xns**, and **x25**. |
| **high** \| **medium** \| **normal** \| **low** | Priority queue level. |
| *queue-keyword keyword-value* | Possible keywords are **fragments**, **gt**, **lt**, **list**, **tcp**, and **udp**. See Table 79. |

## Default

No queuing priorities are established.

## Command Mode

Global configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

When using multiple rules for a single protocol, remember that the system reads the priority settings in order of appearance. When classifying a packet, the system searches the list of rules specified by **priority-list** commands for a matching protocol type. When a match is found, the packet is assigned to the appropriate queue. The list is searched in the order it is specified, and the first matching rule terminates the search.

The **decnet_router-l1** keyword refers to the multicast address for all level-1 routers, which are intra-area routers, and the **decnet_router-l2** keyword refers to all level 2 routers, which are interarea routers.

The **dlsw**, **rsrb**, and **stun** keywords refer only to direct encapsulation.

Use Table 79, Table 80, and Table 81 to configure the queuing priorities for your system.

**Table 79 Protocol Priority Queue Keywords and Values**

| Option | Description |
|---|---|
| **fragments** | Assigns the priority level defined to fragmented IP packets (for use with IP protocol only). More specifically, IP packets whose fragment offset field is nonzero are matched by this command. The initial fragment of a fragmented IP packet has a fragment offset of zero, so such packets are not matched by this command.<br>Note: Packets with a nonzero fragment offset do not contain TCP or UDP headers, so other instances of this command that use the **tcp** or **udp** keyword will always fail to match such packets. |
| **gt** *byte-count* | Specifies a greater-than count. The priority level assigned goes into effect when a packet size exceeds the value entered for the argument *byte-count*. The size of the packet must also include additional bytes because of MAC encapsulation on the outgoing interface. |
| **lt** *byte-count* | Specifies a less-than count. The priority level assigned goes into effect when a packet size is less than the value entered for the argument *byte-count*. The size of the packet must also include additional bytes because of MAC encapsulation on the outgoing interface. |
| **list** *list-number* | Assigns traffic priorities according to a specified list when used with AppleTalk, bridging, IP, IPX, VINES, or XNS. The *list-number* argument is the access list number as specified by the **access-list** global configuration command for the specified *protocol-name*. For example, if the protocol is AppleTalk, *list-number* should be a valid AppleTalk access list number. |
| **tcp** *port* | Assigns the priority level defined to TCP segments originating from or destined to a specified port (for use with the IP protocol only). Table 80 lists common TCP services and their port numbers. |
| **udp** *port* | Assigns the priority level defined to UDP packets originating from or destined to a specified port (for use with the IP protocol only). Table 81 lists common UDP services and their port numbers. |

**Table 80 Common TCP Services and Their Port Numbers**

| Service | Port |
|---|---|
| Telnet | 23 |
| SMTP | 25 |

**Table 81 Common UDP Services and Their Port Numbers**

| Service | Port |
|---|---|
| TFTP | 69 |
| NFS | 2049 |
| SNMP | 161 |
| RPC | 111 |
| DNS | 53 |

**Note** The TCP and UDP ports listed in Table 80 and Table 81 include some of the more common port numbers. However, you can specify any port number to be prioritized; you are not limited to those listed.
For some protocols, such as TFTP and FTP, only the initial request uses port 69. Subsequent packets use a randomly chosen port number. For these types of protocols, the use of port numbers fails to be an effective method to manage queued traffic.

Use the **no priority-list** global configuration command followed by the appropriate *list-number* argument and the **protocol** keyword to remove a priority list entry assigned by protocol type.

## Examples

The following example assigns 1 as the arbitrary priority list number, specifies DECnet as the protocol type, and assigns a high-priority level to the DECnet packets transmitted on this interface:

```
priority-list 1 protocol decnet high
```

The following example assigns a medium-priority level to every DECnet packet with a size greater than 200 bytes:

```
priority-list 2 protocol decnet medium gt 200
```

The following example assigns a medium-priority level to every DECnet packet with a size less than 200 bytes:

```
priority-list 4 protocol decnet medium lt 200
```

The following example assigns a high-priority level to traffic that matches IP access list 10:

```
priority-list 1 protocol ip high list 10
```

The following example assigns a medium-priority level to Telnet packets:

```
priority-list 4 protocol ip medium tcp 23
```

The following example assigns a medium-priority level to UDP Domain Name service packets:

```
priority-list 4 protocol ip medium udp 53
```

The following example assigns a high-priority level to traffic that matches Ethernet type code access list 201:

```
priority-list 1 protocol bridge high list 201
```

The following example assigns a high-priority level to DLSw+ traffic with TCP encapsulation:

```
priority-list 1 protocol ip  high  tcp 2065
```

The following example assigns a high-priority level to DLSw+ traffic with Direct encapsulation:

```
priority-list 1 protocol dlsw high
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**priority-group**
**show queueing**

# priority-list queue-limit

To specify the maximum number of packets that can be waiting in each of the priority queues, use the **priority-list queue-limit** global configuration command. The **no** form of this command selects the normal queue.

**priority-list** *list-number* **queue-limit** *high-limit medium-limit normal-limit low-limit*
**no priority-list** *list-number* **queue-limit**

## Syntax Description

| | |
|---|---|
| *list-number* | Arbitrary integer between 1 and 16 that identifies the priority list selected by the user. |
| *high-limit medium-limit normal-limit low-limit* | Priority queue maximum length. A value of 0 for any of the four arguments means that the queue can be of unlimited size for that particular queue. |

## Default

The default queue limit arguments are listed in Table 82.

**Table 82 Priority Queue Packet Limits**

| Priority Queue Argument | Packet Limits |
|---|---|
| *high-limit* | 20 |
| *medium-limit* | 40 |
| *normal-limit* | 60 |
| *low-limit* | 80 |

## Command Mode

Global configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

If a priority queue overflows, excess packets are discarded and quench messages can be sent, if appropriate, for the protocol.

## Example

The following example sets the maximum packets in the priority queue to 10:

```
priority-list 2 queue-limit 10 40 60 80
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**priority-group**
**show queueing**

# queue-list default

To assign a priority queue for those packets that do not match any other rule in the queue list, use the **queue-list default** global configuration command. To restore the default value, use the **no** form of this command.

**queue-list** *list-number* **default** *queue-number*
**no queue-list** *list-number* **default** *queue-number*

## Syntax Description

| | |
|---|---|
| *list-number* | Number of the queue list. An integer from 1 to 16. |
| *queue-number* | Number of the queue. An integer from 1 to 16. |

## Default

Queue number 1

## Command Mode

Global configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

When using multiple rules, remember that the system reads the **queue-list** commands in order of appearance. When classifying a packet, the system searches the list of rules specified by **queue-list** commands for a matching protocol or interface type. When a match is found, the packet is assigned to the appropriate queue. The list is searched in the order it is specified, and the first matching rule terminates the search.

Queue number 0 is a system queue. It is emptied before any of the other queues are processed. The system enqueues high-priority packets, such as keepalives, to this queue.

Use the **show interface** command to display the current status of the output queues.

## Example

In the following example, the default queue for list 10 is set to queue number 2:

```
queue-list 10 default 2
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**custom-queue-list**
**show queueing**

# queue-list interface

To establish queuing priorities on packets entering on an interface, use the **queue-list interface** global configuration command. To remove an entry from the list, use the **no** form of the command.

**queue-list** *list-number* **interface** *type number queue-number*
**no queue-list** *list-number* **interface** *queue-number*

## Syntax Description

| | |
|---|---|
| *list-number* | Number of the queue list. An integer from 1 to 16. |
| *type* | Required argument that specifies the name of the interface. |
| *number* | Number of the specified interface. |
| *queue-number* | Number of the queue. An integer from 1 to 16. |

## Default

No queuing priorities are established.

## Command Mode

Global configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

When using multiple rules, remember that the system reads the **queue-list** commands in order of appearance. When classifying a packet, the system searches the list of rules specified by **queue-list** commands for a matching protocol or interface type. When a match is found, the packet is assigned to the appropriate queue. The list is searched in the order it is specified, and the first matching rule terminates the search.

## Example

In the following example, queue list 4 established queuing priorities for packets entering on interface tunnel 3. The queue number assigned is 10.

```
queue-list 4 interface tunnel 3 10
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**custom-queue-list**
**show queueing**

# queue-list protocol

To establish queuing priority based upon the protocol type, use the **queue-list protocol** global configuration command. Use the **no** form of this command with the appropriate list number to remove an entry from the list.

**queue-list** *list-number* **protocol** *protocol-name queue-number queue-keyword keyword-value*
**no queue-list** *list-number* **protocol** *protocol-name*

## Syntax Description

| | |
|---|---|
| *list-number* | Number of the queue list. An integer from 1 to 16. |
| *protocol-name* | Required argument that specifies the protocol type: **aarp**, **arp**, **apollo**, **appletalk**, **bridge** (transparent), **clns**, **clns_es**, **clns_is**, **compressedtcp**, **cmns**, **decnet**, **decnet_node**, **decnet_routerl1**, **decnet_routerl2**, **dlsw**, **ip**, **ipx**, **pad**, **rsrb**, **stun**, **vines**, **xns**, and **x25**. |
| *queue-number* | Number of the queue. An integer from 1 to 16. |
| *queue-keyword keyword-value* | Possible keywords are **gt**, **lt**, **list**, **tcp**, and **udp**. See Table 79. |

## Default

No queuing priorities are established.

## Command Mode

Global configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

When using multiple rules, remember that the system reads the **queue-list** commands in order of appearance. When classifying a packet, the system searches the list of rules specified by **queue-list** commands for a matching protocol or interface type. When a match is found, the packet is assigned to the appropriate queue. The list is searched in the order it is specified, and the first matching rule terminates the search.

The **decnet_router-l1** keyword refers to the multicast address for all level-1 routers, which are intra-area routers, and the **decnet_router-l2** keyword refers to all level 2 routers, which are interarea routers.

The **dlsw**, **rsrb**, and **stun** keywords refer only to direct encapsulation.

Use Table 79, Table 80, and Table 81 from the **priority-list protocol** command to configure custom queueing for your system.

## Examples

The following example assigns 1 as the custom queue list, specifies DECnet as the protocol type, and assigns 3 as a queue number to the packets transmitted on this interface:

```
queue-list 1 protocol decnet 3
```

The following example assigns DECnet packets with a size greater than 200 bytes to queue number 2:

```
queue-list 2 protocol decnet 2 gt 200
```

The following example assigns DECnet packets with a size less than 200 bytes to queue number 2:

```
queue-list 4 protocol decnet 2 lt 200
```

The following example assigns traffic that matches IP access list 10 to queue number 1:

```
queue-list 1 protocol ip 1 list 10
```

The following example assigns Telnet packets to queue number 2:

```
queue-list 4 protocol ip 2 tcp 23
```

The following example assigns UDP Domain Name service packets to queue number 2:

```
queue-list 4 protocol ip 2 udp 53
```

The following example assigns traffic that matches Ethernet type code access list 201 to queue number 1:

```
queue-list 1 protocol bridge 1 list 201
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**custom-queue-list**
**show queueing**

# queue-list queue byte-count

To designate the byte size allowed per queue, use the **queue-list queue byte-count** global configuration command. To return the byte size to the default value, use the **no** form of the command.

**queue-list** *list-number* **queue** *queue-number* **byte-count** *byte-count-number*
**no queue-list** *list-number* **queue** *queue-number* **byte-count** *byte-count-number*

## Syntax Description

| | |
|---|---|
| *list-number* | Number of the queue list. An integer from 1 to 16. |
| *queue-number* | Number of the queue. An integer from 1 to 16. |
| *byte-count-number* | Specifies the lower boundary on how many bytes the system allows to be delivered from a given queue during a particular cycle. |

## Default

1500 bytes

## Command Mode

Global configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

## Example

In the following example, queue list 9 establishes the byte-count as 1400 for queue number 10:

```
queue-list 9 queue 10 byte-count 1400
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**custom-queue-list**
**show queueing**

# queue-list queue limit

To designate the queue length limit for a queue, use the **queue-list queue limit** global configuration command. To return the queue length to the default value, use the **no** form of the command.

**queue-list** *list-number* **queue** *queue-number* **limit** *limit-number*
**no queue-list** *list-number* **queue** *queue-number* **limit** *limit-number*

## Syntax Description

| | |
|---|---|
| *list-number* | Number of the queue list. An integer from 1 to 16. |
| *queue-number* | Number of the queue. An integer from 1 to 16. |
| *limit-number* | Maximum number of packets which can be enqueued at any time. Range is 0 to 32767 queue entries. A value of 0 means that the queue can be of unlimited size. |

## Default

20 entries

## Command Mode

Global configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

## Example

In the following example, the queue length of queue 10 is increased to 40:

```
queue-list 5 queue 10 limit 40
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**custom-queue-list**
**show queueing**

# random-detect

To enable random early detection on an interface, use the **random-detect** interface configuration command. Use the **no** form of this command to disable random early detection on the interface.

**random-detect** [*weighting*]
**no random-detect**

## Syntax Description

| | |
|---|---|
| *weighting* | (Optional) Exponential weighting constant in the range 1 to 16 used to determine the rate that packets are dropped when congestion occurs. The default is 10 (that is, drop 1 packet every $2^{10}$). |

## Default

Random early detection is disabled.

## Command Mode

Interface configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Random early detection (RED) is useful in high-speed TCP/IP networks to avoid congestion by dropping packets at a controlled rate. RED is not recommended for protocols, such as AppleTalk or Novell Netware, that respond to dropped packets by retransmitting the packets at the same rate. RED should only be configured on an interface where most of the traffic is TCP/IP traffic.

Cisco recommends using the default value for the exponential weighting constant; however, you may need to change this value depending on your operational environment. For example, a value of 10 (the default), which might achieve a loss rate of $10^{-4}$, is recommended for high-speed links such as DS3 and OC3, whereas a value of 7, which might achieve a loss rate of $10^{-3}$, is recommended for T1 links.

Random early detection cannot be configured on an interface already configured with custom, priority, or fair queueing.

When RSVP is configured on the interface, packets from other traffic flows are dropped before RSVP flows (when possible). Also, the IP precedence of the packet determines whether the packet is dropped. Lower-precedence traffic is dropped before higher-precedence traffic. Therefore, lower-precedence traffic is more likely to be throttled back.

## Example

The following example enables random early detection on a serial interface:

```
interface serial 0
random-detect
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**custom-queue-list**
**fair-queue**
**ip rsvp bandwidth**
**priority-group**
**priority-list default**
**queue-list default**
**random-detect**
**show interface**

# scheduler allocate

To guarantee CPU time for processes, use the **scheduler allocate** global configuration command on the Cisco 7200 series and Cisco 7500 series. The **no** form of this command restores the default.

**scheduler allocate** *interrupt-time process-time*
**no scheduler allocate**

## Syntax Description

| | |
|---|---|
| *interrupt-time* | Integer (in microseconds) that limits the maximum number of microseconds to spend on fast switching within any one network interrupt context. The range is 400 to 60000 microseconds. The default is 4000 microseconds. |
| *process-time* | Integer (in microseconds) that guarantees the minimum number of microseconds to spend at the process level when network interrupts are disabled. The range is 100 to 4000. The default is 200 microseconds. |

## Default

Approximately 5 percent of the CPU is available for process tasks.

## Command Mode

Global configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

This command applies to the Cisco 7200 series and Cisco 7500 series.

**Caution** Cisco recommends that you do not change the default values.

## Example

The following example makes 20 percent of the CPU available for process tasks:

```
scheduler allocate 2000 500
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**scheduler interval**
**scheduler process-watchdog**

# scheduler interval

To control the maximum amount of time that can elapse without running system processes, use the **scheduler interval** global configuration command. The **no** form of this command restores the default.

**scheduler interval** *milliseconds*
**no scheduler interval**

## Syntax Description

| | |
|---|---|
| *milliseconds* | Integer that specifies the interval, in milliseconds. The minimum interval that you can specify is 500 milliseconds; there is no maximum value. |

## Default

High-priority operations are allowed to use as much of the central processor as needed.

## Command Mode

Global configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

The normal operation of the network server allows the switching operations to use as much of the central processor as is required. If the network is running unusually heavy loads that do not allow the processor the time to handle the routing protocols, give priority to the system process scheduler. High-priority operations are allowed to use as much of the central processor as needed.

On the Cisco 7200 series and Cisco 7500 series, use the **scheduler allocate** global configuration command.

## Example

The following example changes the low-priority process schedule to an interval of 750 milliseconds:

```
scheduler interval 750
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**scheduler allocate**
**scheduler process-watchdog**

# scheduler process-watchdog

Use the **scheduler process-watchdog** global configuration command to configure the characteristics for a looping process. Use the **no** form of this command to disable the looping configuration.

**scheduler process-watchdog**{**hang** | **normal** | **reload** | **terminate**}
**no scheduler process-watchdog**{**hang** | **normal** | **reload** | **terminate**}

## Syntax Description

| | |
|---|---|
| **hang** | Retains the process, but does not schedule it. |
| **normal** | Indicates the factory specified per-process behavior. |
| **reload** | Reloads the system. |
| **terminate** | Terminates the process and continues. |

## Command Mode

Global configuration

## Usage Guidelines

This command was introduced in a release prior to Cisco IOS Release 11.3.

This command applies to the Cisco 7200 series and Cisco 7500 series.

Example

In the following example, the looping processes are configured to follow the factory specified per-process behavior.

```
scheduler process-watchdog normal
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**scheduler allocate**
**scheduler interval**

# service nagle

To enable the Nagle congestion control algorithm, use the **service nagle** global configuration command. Use the **no** form of this command to disable this feature.

**service nagle**
**no service nagle**

## Syntax Description

This command has no arguments or keywords.

## Default

Disabled

## Command Mode

Global configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

When using a standard TCP implementation to send keystrokes between machines, TCP tends to send one packet for each keystroke typed. On larger networks, many small packets use up bandwidth and contribute to congestion.

John Nagle's algorithm (RFC 896) helps alleviate the small-packet problem in TCP. In general, it works this way: The first character typed after connection establishment is sent in a single packet, but TCP holds any additional characters typed until the receiver acknowledges the previous packet. Then the second, larger packet is sent, and additional typed characters are saved until the acknowledgment comes back. The effect is to accumulate characters into larger chunks, and pace them out to the network at a rate matching the round-trip time of the given connection. This method is usually a good for all TCP-based traffic. However, do not use the **service nagle** command if you have XRemote users on X Window sessions.

## Example

The following example enables the Nagle algorithm:

```
service nagle
```

# show buffers

To display statistics for the buffer pools on the network server, use the **show buffers** EXEC command.

**show buffers** [**address** *hex-addr* |[**all** | **assigned** | **failures** | **free** | **old** [**dump** | **header** | **packet**]] | **input-interface** *interface-type identifier* | **pool** *pool-name*]

## Syntax Description

| | |
|---|---|
| **address** | Displays buffers at a specified address. |
| *hex-addr* | Address, in hexadecimal notation, of the buffer to display. |
| **all** | Displays all buffers. |
| **assigned** | Displays the buffers in use. |
| **failures** | Displays buffer allocation failures. |
| **free** | Displays the buffers available for use. |
| **old** | Displays buffers older than one minute. |
| **dump** | Shows the buffer header and all data in the display. |
| **header** | Shows only the buffer header in the display. |
| **packet** | Shows the buffer header and packet data in the display. |
| **input-interface** | Displays interface pool information. If the specified *interface-type* has its own buffer pool, displays information for that pool. |
| *interface-type* | Specifies an input interface as **ethernet**, **fastethernet**, **loopback**, **serial**, or **null**. |
| *identifier* | Identifier of the interface specified in *interface-type*. |
| **pool** | Displays buffers in a specified buffer pool. |
| *pool-name* | Specifies the name of a buffer pool to use. |

## Command Mode

EXEC

## Default

The **show buffers** command without any arguments displays all buffer pool information

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

## Sample Displays

The following is sample output from the **show buffers** command with no arguments, showing all buffer pool information:

```
Router# show buffers

Buffer elements:
     398 in free list (500 max allowed)
     1266 hits, 0 misses, 0 created

Public buffer pools:
Small buffers, 104 bytes (total 50, permanent 50):
     50 in free list (20 min, 150 max allowed)
     551 hits, 0 misses, 0 trims, 0 created
Middle buffers, 600 bytes (total 25, permanent 25):
     25 in free list (10 min, 150 max allowed)
     39 hits, 0 misses, 0 trims, 0 created
Big buffers, 1524 bytes (total 50, permanent 50):
     49 in free list (5 min, 150 max allowed)
     27 hits, 0 misses, 0 trims, 0 created
VeryBig buffers, 4520 bytes (total 10, permanent 10):
     10 in free list (0 min, 100 max allowed)
     0 hits, 0 misses, 0 trims, 0 created
Large buffers, 5024 bytes (total 0, permanent 0):
     0 in free list (0 min, 10 max allowed)
     0 hits, 0 misses, 0 trims, 0 created
Huge buffers, 18024 bytes (total 0, permanent 0):
     0 in free list (0 min, 4 max allowed)
     0 hits, 0 misses, 0 trims, 0 created

Interface buffer pools:
Ethernet0 buffers, 1524 bytes (total 64, permanent 64):
     16 in free list (0 min, 64 max allowed)
     48 hits, 0 fallbacks
     16 max cache size, 16 in cache
Ethernet1 buffers, 1524 bytes (total 64, permanent 64):
     16 in free list (0 min, 64 max allowed)
     48 hits, 0 fallbacks
     16 max cache size, 16 in cache
Serial0 buffers, 1524 bytes (total 64, permanent 64):
     16 in free list (0 min, 64 max allowed)
     48 hits, 0 fallbacks
     16 max cache size, 16 in cache
Serial1 buffers, 1524 bytes (total 64, permanent 64):
     16 in free list (0 min, 64 max allowed)
     48 hits, 0 fallbacks
     16 max cache size, 16 in cache
TokenRing0 buffers, 4516 bytes (total 48, permanent 48):
     0 in free list (0 min, 48 max allowed)
     48 hits, 0 fallbacks
     16 max cache size, 16 in cache
TokenRing1 buffers, 4516 bytes (total 32, permanent 32):
     32 in free list (0 min, 48 max allowed)
     16 hits, 0 fallbacks

0 failures (0 no memory)
```

Table 83 describes significant fields shown in the display.

**Table 83 Show Buffers Field Descriptions**

| Field | Description |
|---|---|
| Buffer elements | Buffer elements are small structures used as placeholders for buffers in internal operating system queues. Buffer elements are used when a buffer may need to be on more than one queue. |
| free list | Total number of the currently unallocated buffer elements. |
| max allowed | Maximum number of buffers that are available for allocation. |
| hits | Count of successful attempts to allocate a buffer when needed. |
| misses | Count of buffer allocation attempts that resulted in growing the buffer pool to allocate a buffer. |
| created | Count of new buffers created to satisfy buffer allocation attempts when the available buffers in the pool have already been allocated. |
| Public buffer pools: | |
| Small buffers | Buffers that are 104 bytes long. |
| Middle buffers | Buffers that are 600 bytes long. |
| Big buffers | Buffers that are 1524 bytes long. |
| VeryBig buffers | Buffers that are 4520 bytes long. |
| Large buffers | Buffers that are 5024 bytes long. |
| Huge buffers | Buffers that are 18024 bytes long. |
| total | Total number of this type of buffer. |
| permanent | Number of these buffers that are permanent. |
| free list | Number of available or unallocated buffers in that pool. |
| min | Minimum number of free or unallocated buffers in the buffer pool |
| max allowed | Maximum number of free or unallocated buffers in the buffer pool |
| hits | Count of successful attempts to allocate a buffer when needed. |
| misses | Count of buffer allocation attempts that resulted in growing the buffer pool in order to allocate a buffer. |
| trims | Count of buffers released to the system because they were not being used. This field is displayed only for dynamic buffer pools, not interface buffer pools, which are static. |
| created | Count of new buffers created in response to misses. This field is displayed only for dynamic buffer pools, not interface buffer pools, which are static. |
| Interface buffer pools: | |
| total | Total number of this type of buffer. |
| permanent | Number of these buffers that are permanent. |
| free list | Number of available or unallocated buffers in that pool. |
| min | Minimum number of free or unallocated buffers in the buffer pool. |
| max allowed | Maximum number of free or unallocated buffers in the buffer pool. |
| hits | Count of successful attempts to allocate a buffer when needed. |
| fallbacks | Count of buffer allocation attempts that resulted in falling back to the public buffer pool that is the smallest pool at least as big as the interface buffer pool. |

**Table 83 Show Buffers Field Descriptions (Continued)**

| Field | Description |
|---|---|
| max cache size | Maximum number of buffers from that interface's pool that can be in that interface buffer pool's cache. Each interface buffer pool has its own cache. These are not additional to the permanent buffers; they come from the interface's buffer pools. Some interfaces place all of their buffers from the interface pool into the cache. In this case, it is normal for the *free list* to display 0. |
| failures | Total number of allocation requests that have failed because no buffer was available for allocation; the datagram was lost. Such failures normally occur at interrupt level. |
| no memory | Number of failures that occurred because no memory was available to create a new buffer. |

The following is sample output from the **show buffers** command with an interface *type* and *identifier*:

```
Router# show buffers Ethernet 0

Ethernet0 buffers, 1524 bytes (total 64, permanent 64):
     16 in free list (0 min, 64 max allowed)
     48 hits, 0 fallbacks
     16 max cache size, 16 in cache
```

# show queueing

To list the current state of the queue lists, use the **show queueing** privileged EXEC command.

**show queueing** [**custom** | **fair** | **priority** | **virtual-access** *interface-number*]

## Syntax Description

| | |
|---|---|
| **custom** | (Optional) Shows status of custom queueing list configuration. |
| **fair** | (Optional) Shows status of the fair queueing configuration. This is the default. |
| **priority** | (Optional) Shows status of priority queueing list configuration. |
| **virtual-access** *interface- number* | (Optional) Shows information about interleaving on a virtual access interface. |

## Default

Fair queueing configuration

## Command Mode

Privileged EXEC

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.3.

If no keyword is entered, this command show the status of fair queueing configuration.

## Sample Displays

The following is sample output from the **show queueing custom** EXEC command:

```
Router# show queueing custom

Current custom queue configuration:

List   Queue  Args
3      10     default
3      3      interface Tunnel3
3      3      protocol ip
3      3      byte-count 444 limit 3
```

The following is sample output from the **show queueing** command. On interface Serial0, there are two active conversations. Weighted fair queueing ensures that both of these IP data streams—both using TCP—receive equal bandwidth on the interface while they have messages in the pipeline, even though there is more FTP data in the queue than rcp data.

```
Router# show queueing

Current fair queue configuration:
Interface Serial0
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Output queue: 18/64/30 (size/threshold/drops)
```

```
   Conversations 2/8 (active/max active)
   Reserved Conversations 0/0 (allocated/max allocated)

(depth/weight/discards) 3/4096/30
Conversation 117, linktype: ip, length: 556, flags: 0x280
source: 172.31.128.115, destination: 172.31.58.89, id: 0x1069, ttl: 59,
TOS: 0 prot: 6, source port 514, destination port 1022

(depth/weight/discards) 14/4096/0
Conversation 155, linktype: ip, length: 1504, flags: 0x280
source: 172.31.128.115, destination: 172.31.58.89, id: 0x104D, ttl: 59,
TOS: 0 prot: 6, source port 20, destination port 1554
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**custom-queue-list**
**priority-group**
**priority-list interface**
**priority-list queue-limit**
**queue-list default**
**queue-list default**
**queue-list interface**
**queue-list protocol**
**queue-list queue byte-count**
**queue-list queue limit**

# show traffic-shape

Use the **show traffic-shape** EXEC command to display the current traffic-shaping configuration.

**show traffic-shape** [*interface*]

## Syntax Description

| | |
|---|---|
| *interface* | (Optional) Name of the interface. |

## Command Mode

EXEC

## Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

You must have first enabled traffic shaping using the **traffic-shape rate**, **traffic-shape group**, or **frame-relay traffic-shaping** command to display traffic-shaping information with the **show traffic-shape** command.

## Sample Display

The following is sample output from the **show traffic-shape** command.

```
Router# show traffic-shape

          access Target    Byte   Sustain   Excess    Interval  Increment Adapt
I/F       list   Rate      Limit  bits/int  bits/int  (ms)       (bytes)  Active
Et0       101    1000000   23437  125000    125000    63        7813      -
Et1              5000000   87889  625000    625000    16        9766      -
```

Table 84 describes the fields shown in the display.

**Table 84 Show Traffic-Shape Field Descriptions**

| Field | Description |
|---|---|
| I/F | Interface. |
| access list | Number of the access list. |
| Target Rate | Rate that traffic is shaped to in bps. |
| Byte Limit | Maximum number of bytes transmitted per internal interval. |
| Sustain bits/int | Configured sustained bits per interval. |
| Excess bits/int | Configured excess bits in the first interval. |
| Interval (ms) | Interval being used internally. This interval may be smaller than the Committed Burst divided by the committed information rate if the router determines that traffic flow will be more stable with a smaller configured interval. |
| Increment (bytes) | Number of bytes that will be sustained per internal interval. |
| Adapt Active | Contains “BECN” if Frame Relay has BECN Adaptation configured. |

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**frame-relay traffic-shaping**
**show traffic-shape statistics**
**traffic-shape adaptive**
**traffic-shape group**
**traffic-shape rate**
**traffic-shape fecn-adapt**

# show traffic-shape statistics

Use the **show traffic-shape statistics** EXEC command to display the current traffic-shaping statistics.

**show traffic-shape statistics** [*interface*]

## Syntax Description

| | |
|---|---|
| *interface* | (Optional) Name of the interface. |

## Command Mode

EXEC

## Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

You must have first enabled traffic shaping using the **traffic-shape rate**, **traffic-shape group**, or **frame-relay traffic-shaping** command to display traffic-shaping information with the **show traffic-shape statistics** command.

## Sample Display

The following is sample output from the **show traffic-shape statistics** command.

```
Router# show traffic-shape statistics

          Access Queue     Packets   Bytes     Packets   Bytes     Shaping
I/F       List   Depth                        Delayed   Delayed   Active
Et0       101    0         2         180       0         0         no
Et1              0         0         0         0         0         no
```

Table 85 describes the fields shown in the display.

**Table 85 Show Traffic-Shape Statistics Field Descriptions**

| Field | Description |
|---|---|
| I/F | Interface. |
| Access List | Number of the access list. |
| Queue Depth | Number of messages in the queue. |
| Packets | Number of packets sent through the interface. |
| Bytes | Number of bytes sent through the interface. |
| Packets Delayed | Number of packets sent through the interface that were delayed in the traffic shaping queue. |
| Bytes Delayed | Number of bytes sent through the interface that were delayed in the traffic shaping queue. |
| Shaping Active | Contains “yes” when timers indicate that traffic shaping is occurring and “no” if traffic shaping is not occurring. |

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**frame-relay traffic-shaping**
**show traffic-shape**
**traffic-shape adaptive**
**traffic-shape group**
**traffic-shape rate**
**traffic-shape fecn-adapt**

# traffic-shape adaptive

To configure a Frame Relay subinterface to estimate the available bandwidth when backward explicit congestion notifications (BECNs) are received, use the **traffic-shape adaptive** interface configuration command. Use the **no** form of this command to stop adapting to congestion signals.

**traffic-shape adaptive** [*bit-rate*]
**no traffic-shape adaptive**

## Syntax Description

| | |
|---|---|
| *bit-rate* | (Optional) Lowest bit rate that traffic is shaped to in bits per second. The default is half the value specified for the **traffic-shape rate** or **traffic-shape group** *bit-rate* option. |

## Default

No available bandwidth is estimated when BECNs are received.

## Command Mode

Interface configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

You must enable traffic shaping on the interface with the **traffic-shape rate** command before you can use the **traffic-shape adaptive** command.

The bit rate specified for the **traffic-shape rate** command is the upper limit, and the bit rate specified for the **traffic-shape adaptive** command is the lower limit to which traffic is shaped when BECNs are received on the interface. The rate actually shaped to will be between these two rates. The **traffic-shape adaptive** command should be configured at both ends of the link, as it also configures the device at the flow end to reflect forward explicit congestion notification (FECN) signals as BECNs, enabling the router at the high-speed end to detect and adapt to congestion even when traffic is flowing primarily in one direction.

## Example

The following example configures traffic shaping on interface 0.1 with an upper limit of 128 kbps and a lower limit of 64 kbps. This allows the link to run from 64 to 128 kbps, depending on the congestion level.

```
interface serial 0
 encapsulation-frame-relay
interface serial 0.1
 traffic-shape rate 128000
 traffic-shape adaptive 64000
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**traffic-shape group**
**traffic-shape rate**
**traffic-shape fecn-adapt**

# traffic-shape fecn-adapt

To reply to messages with the forward explicit congestion notification (FECN) bit, use the **traffic-shape fecn-adapt** interface configuration command which generates TEST RESPONSE messages with the BECN bit set. To stop backward explicit congestion notification (BECN) message generation, use the **no** form of this command.

**traffic-shape fecn-adapt**
**no traffic-shape fecn-adapt**

## Syntax Description

This command has no arguments or keywords.

## Defaults

Traffic shaping is disabled.

This command is only available on routers with the Frame Relay interface.

## Command Modes

Interface configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Enable traffic shaping on the interface with the **traffic-shape rate** or **traffic-shape group** command. The **traffic-shape fecn-adapt** command is available only when traffic shaping is configured.

Use this command to reflect FECN bits as BECN bits to notify the other data terminal equipment (DTE) that it is transmitting too fast to slow down and eventually adapt to the minimum bit rate. Use the **traffic-shape adaptive** command to configure the router to adapt its transmission rate when it receives BECNs.

## Examples

The following example configures two routers connected through the Frame Relay network. Router1 is the router on which you want to adapt the network congestion when there is not enough reverse traffic to carry BECNS to this router.

Traffic shaping is configured on both routers on serial interface 0.1 with an upper limit of 128 kbps with the **traffic-shape rate** command. The **traffic-shape adaptive** command is used to configure router1 at 64kbps. This configuration allows the link to run from 64 to 128 kbps, depending on the congestion level. The **traffic-shape fecn-adapt** command is configured on router 2 to reflect FECN bits as BECN bits.

Enter the following on router1:

```
router1(config)# interface serial 0
router1(config-if)# encapsulation-frame-relay
router1(config)# interface serial 0.1
router1(config-if)# traffic-shape rate 128000
router1(config-if)# traffic-shape adaptive 64000
```

Enter the following on router2:

```
router2(config)# interface serial 0
router2(config-if)# encapsulation-frame-relay
router2(config)# interface serial 0.1
router2(config-if)# traffic-shape rate 128000
router2(config-if)# traffic-shape fecn-adapt
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**show traffic-shape**
**show traffic-shape statistics**
**traffic-shape adaptive**
**traffic-shape group**
**traffic-shape rate**

# traffic-shape group

To enable traffic shaping based on a specific access list for outbound traffic on an interface, use the **traffic-shape group** interface configuration command. Use the **no** form of this command to disable traffic shaping on the interface for the access list.

**traffic-shape group** *access-list bit-rate* [*burst-size* [*excess-burst-size*]]
**no traffic-shape group** *access-list*

## Syntax Description

| | |
|---|---|
| *access-list* | Number of the access list that controls the packets that traffic shaping is applied to on the interface. |
| *bit-rate* | Bit rate that traffic is shaped to in bits per second. This is the access bit rate that you contract with your service provider or the service level you intend to maintain. |
| *burst-size* | (Optional) Sustained number of bits that can be transmitted per interval. On Frame Relay interfaces, this is the committed burst size contracted with your service provider. The default is the *bit-rate* divided by 8. |
| *excess-burst-size* | (Optional) Maximum number of bits that can exceed the burst size in the first interval in a congestion event. On Frame Relay interfaces, this is the excess burst size contracted with your service provider. The default is equal to the *burst-size*. |

## Default

Traffic shaping is disabled.

## Command Mode

Interface configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

**Note** Traffic shaping is not supported with optimum, distributed, or flow switching. If you enable this command, all interfaces will revert to fast switching.

Traffic shaping uses queues to limit surges that can congest a network. Data is buffered and then sent into the network in regulated amounts to ensure that traffic will fit within the promised traffic envelope for the particular connection.

The **traffic-shape group** command allows you to specify one or more previously defined access list to shape traffic to on the interface. You must specify one **traffic-shape group** command for each access list on the interface.

You would use traffic shaping if you have a network with differing access rates or if you are offering a subrate service. You can configure the values according to your contract with your service provider or service levels you intend to maintain.

An interval is calculated as follows:

- If the *burst-size* is not equal to zero, the interval is the *burst-size* divided by the *bit-rate*.
- If the *burst-size* is zero, the interval is the *excess-burst-size* divided by the *bit-rate*.

Traffic shaping is supported on all media and encapsulation types on the router. To perform traffic shaping on Frame Relay virtual circuits, you can also use the **frame-relay traffic-shaping** command. For more information on Frame Relay traffic shaping, refer to the "Configuring Frame Relay" chapter in the *Wide-Area Network Configuration Guide*.

If traffic shaping is performed on a Frame Relay network with the **traffic-shape rate** command, you can also use the **traffic-shape adaptive** command to specify the minimum bit-rate the traffic is shaped to.

## Example

The following example enables traffic that matches access list 101 to be shaped to a certain rate and traffic matching access list 102 to be shaped to another rate on the interface:

```
interface serial 1
 traffic-shape rate 128000 16000 8000 group 101
 traffic-shape rate 130000 10000 1000 group 102
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**access-list**
**traffic-shape adaptive**
**traffic-shape rate**
**traffic-shape fecn-adapt**

# traffic-shape rate

To enable traffic shaping for outbound traffic on an interface, use the **traffic-shape rate** interface configuration command. Use the **no** form of this command to disable traffic shaping on the interface.

**traffic-shape rate** *bit-rate* [*burst-size* [*excess-burst-size*]]
**no traffic-shape rate**

## Syntax Description

| | |
|---|---|
| *bit-rate* | Bit rate that traffic is shaped to in bits per second. This is the access bit rate that you contract with your service provider or the service level you intend to maintain. |
| *burst-size* | (Optional) Sustained number of bits that can be transmitted per interval. On Frame Relay interfaces, this is the committed burst size contracted with your service provider. The default is the *bit-rate* divided by 8. |
| *excess-burst-size* | (Optional) Maximum number of bits that can exceed the burst size in the first interval in a congestion event. On Frame Relay interfaces, this is the excess burst size contracted with your service provider. The default is equal to the *burst-size*. |

## Default

Traffic shaping is disabled.

## Command Mode

Interface configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

**Note** Traffic shaping is not supported with optimum, distributed, or flow switching. If you enable this command, all interfaces will revert to fast switching.

Traffic shaping uses queues to limit surges that can congest a network. Data is buffered and then sent into the network in regulated amounts to ensure that traffic will fit within the promised traffic envelope for the particular connection.

You would use traffic shaping if you have a network with differing access rates or if you are offering a subrate service. You can configure the values according to your contract with your service provider or service levels you intend to maintain.

An interval is calculated as follows:

- If the *burst-size* is not equal to zero, the interval is the *burst-size* divided by the *bit-rate*.
- If the *burst-size* is zero, the interval is the *excess-burst-size* divided by the *bit-rate*.

Traffic shaping is supported on all media and encapsulation types on the router. To perform traffic shaping on Frame Relay virtual circuits, you can also use the **frame-relay traffic-shaping** command. For more information on Frame Relay traffic shaping, refer to the "Configuring Frame Relay" chapter in the *Wide-Area Network Configuration Guide*.

If traffic shaping is performed on a Frame Relay network with the **traffic-shape rate** command, you can also use the **traffic-shape adaptive** command to specify the minimum bit-rate the traffic is shaped to.

## Example

The following example enables traffic shaping on a serial interface using the bandwidth required by the service provider:

```
interface serial 0
 traffic-shape rate 128000 16000 8000
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**traffic-shape adaptive**
**traffic-shape group**
**traffic-shape fecn-adapt**